

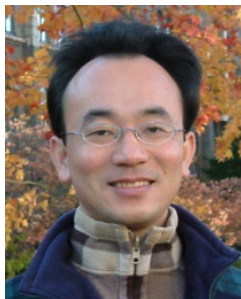
人工智慧文本分析 (AI for Text Analytics)

文本相似度和分群 (Text Similarity and Clustering)

1091AITA08

MBA, IMTKU (M2455) (8418) (Fall 2020)

Thu 3, 4 (10:10-12:00) (B206)



Min-Yuh Day

戴敏育

Associate Professor

副教授

Institute of Information Management, National Taipei University

國立臺北大學 資訊管理研究所

<https://web.ntpu.edu.tw/~myday>

2020-11-26



課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2020/09/17	人工智慧文本分析課程介紹 (Course Orientation on Artificial Intelligence for Text Analytics)
2	2020/09/24	文本分析的基礎：自然語言處理 (Foundations of Text Analytics: Natural Language Processing; NLP)
3	2020/10/01	中秋節 (Mid-Autumn Festival) 放假一天 (Day off)
4	2020/10/08	Python自然語言處理 (Python for Natural Language Processing)
5	2020/10/15	處理和理解文本 (Processing and Understanding Text)
6	2020/10/22	文本表達特徵工程 (Feature Engineering for Text Representation)

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
7	2020/10/29	人工智慧文本分析個案研究 I (Case Study on Artificial Intelligence for Text Analytics I)
8	2020/11/05	文本分類 (Text Classification)
9	2020/11/12	文本摘要和主題模型 (Text Summarization and Topic Models)
10	2020/11/19	期中報告 (Midterm Project Report)
11	2020/11/26	文本相似度和分群 (Text Similarity and Clustering)
12	2020/12/03	語意分析和命名實體識別 (Semantic Analysis and Named Entity Recognition; NER)

課程大綱 (Syllabus)

- | 週次 (Week) | 日期 (Date) | 內容 (Subject/Topics) |
|-----------|------------|--|
| 13 | 2020/12/10 | 情感分析
(Sentiment Analysis) |
| 14 | 2020/12/17 | 人工智慧文本分析個案研究 II
(Case Study on Artificial Intelligence for Text Analytics II) |
| 15 | 2020/12/24 | 深度學習和通用句子嵌入模型
(Deep Learning and Universal Sentence-Embedding Models) |
| 16 | 2020/12/31 | 問答系統與對話系統
(Question Answering and Dialogue Systems) |
| 17 | 2021/01/07 | 期末報告 I (Final Project Presentation I) |
| 18 | 2021/01/14 | 期末報告 II (Final Project Presentation II) |

Outline

- Text Similarity
- Text Clustering
 - Cluster Analysis
 - K-Means Clustering

Text Similarity and Clustering

Text Similarity and Clustering

**Text Dataset
(Unsupervised)**

Text Pre-Processing

**Feature Extraction
(Vectorization) (TF-IDF)(Embedding)**

Text Similarity

Text Clustering

Text Similarity and Clustering

- How do we measure **similarity** between terms and documents?
- How can we use **distance** measures to find the most **relevant documents**?
- How can we build a **recommender system** from **text similarity**?
- How do we **group similar documents** (**document clustering**)?

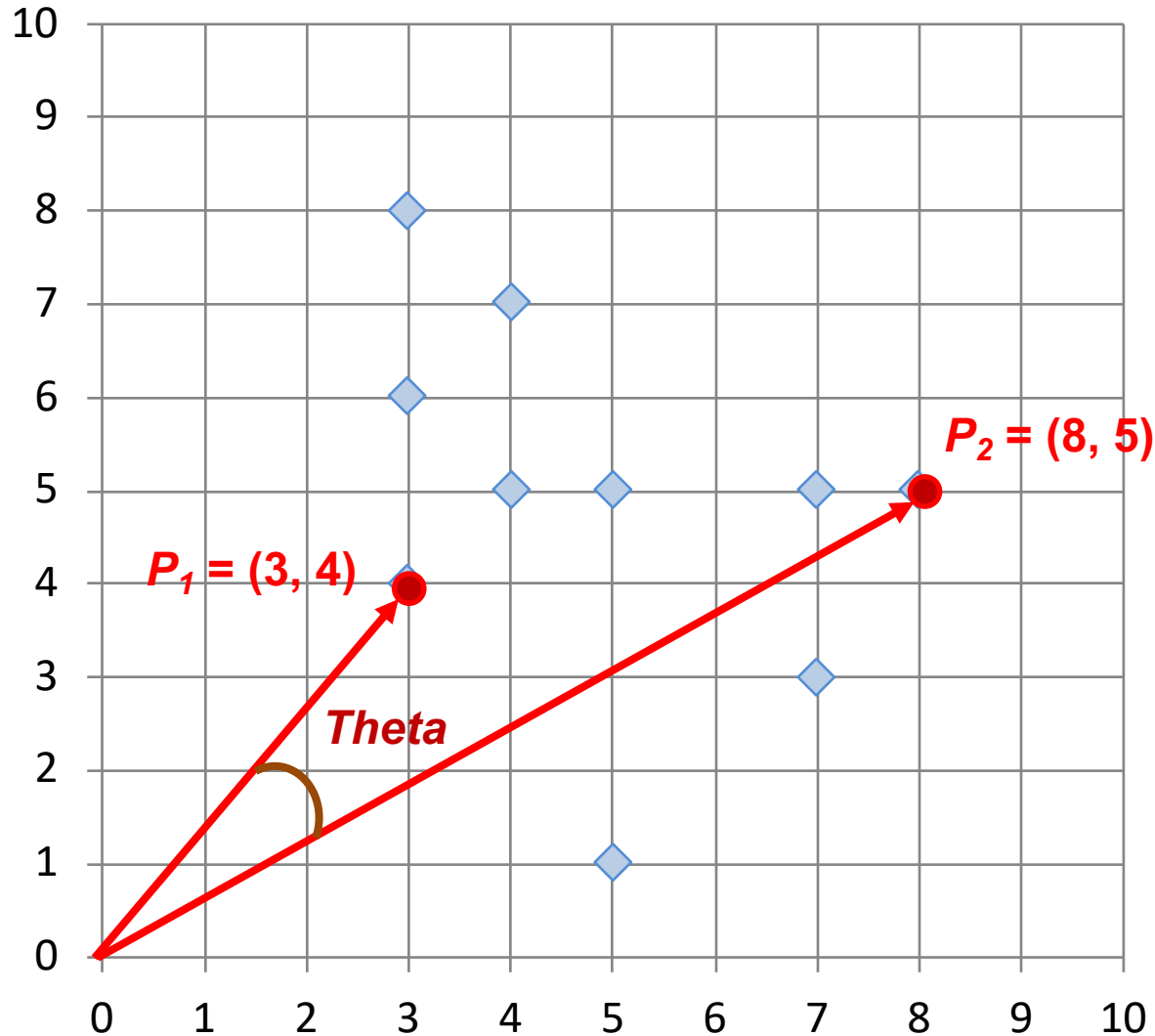
Text Similarity and Clustering

- Information Retrieval (IR)
- Feature Engineering
- Similarity Measures
- Unsupervised Machine Learning Algorithms

Text Similarity

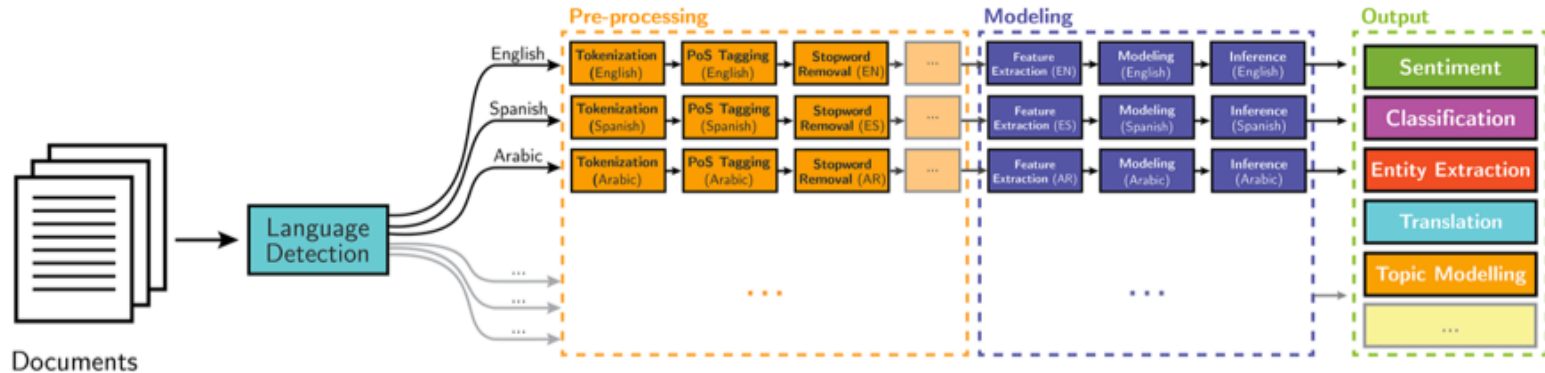
- Lexical similarity
 - Syntax, structure, and content of the documents
- Semantic similarity
 - Semantics, meaning, and context of the documents

Cosine Similarity

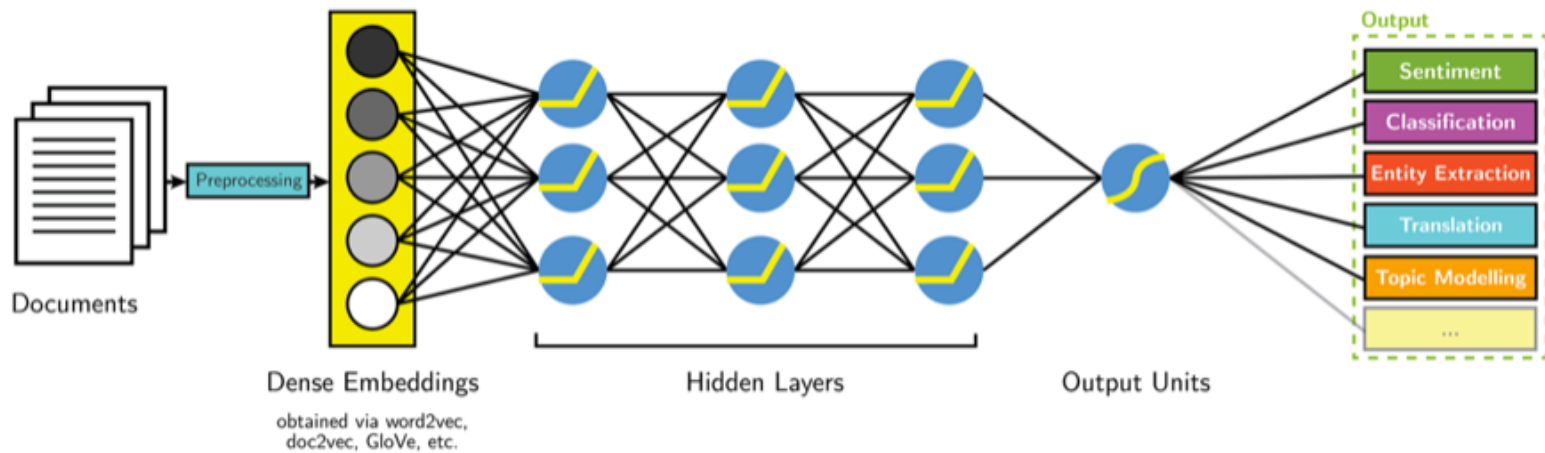


NLP

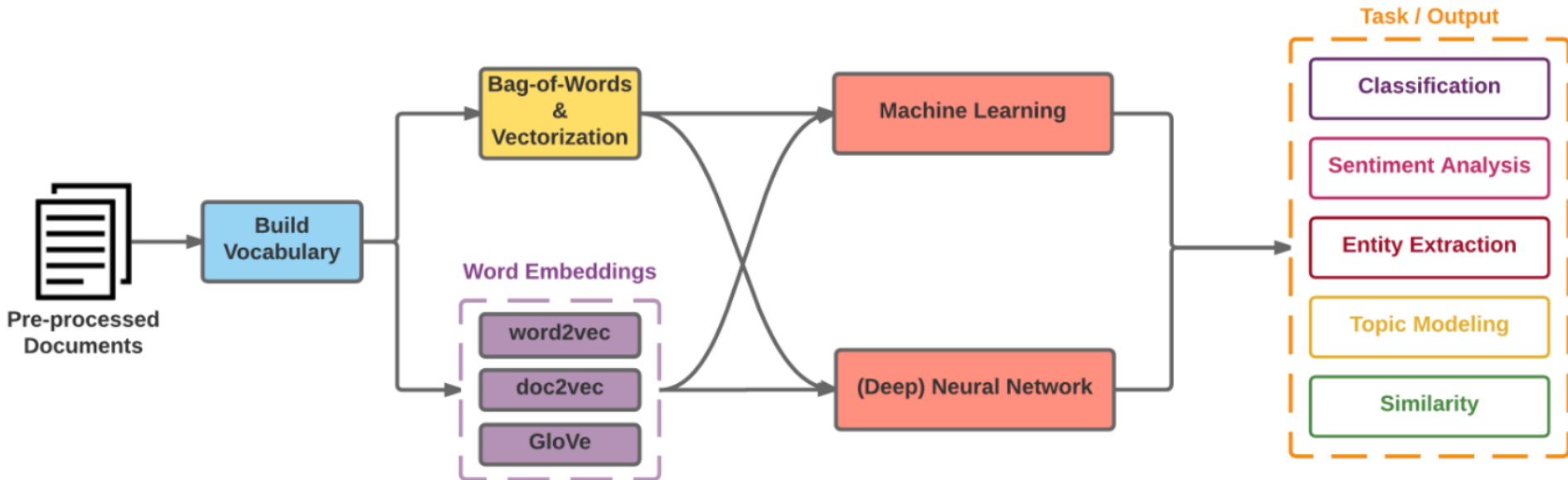
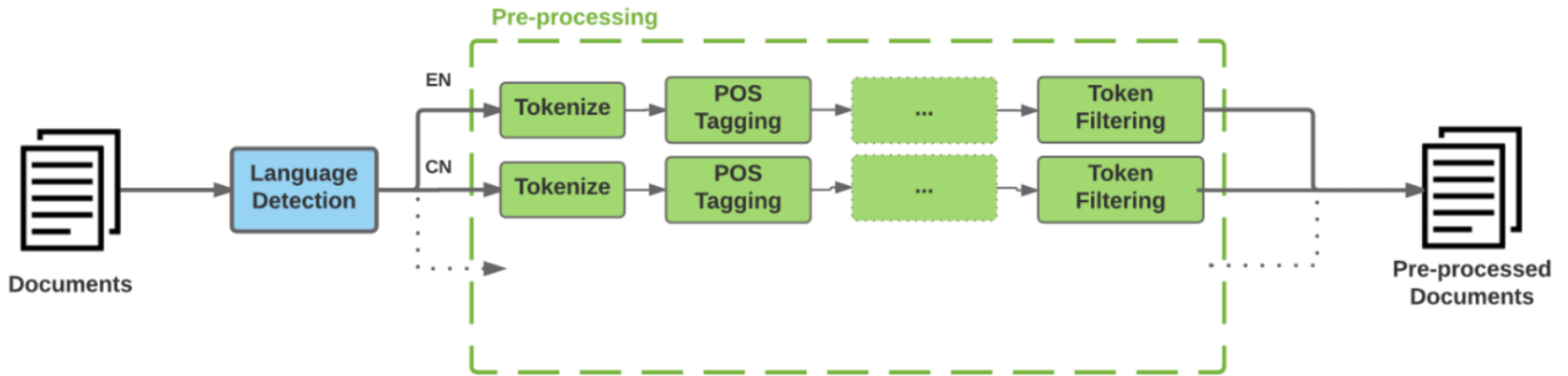
Classical NLP



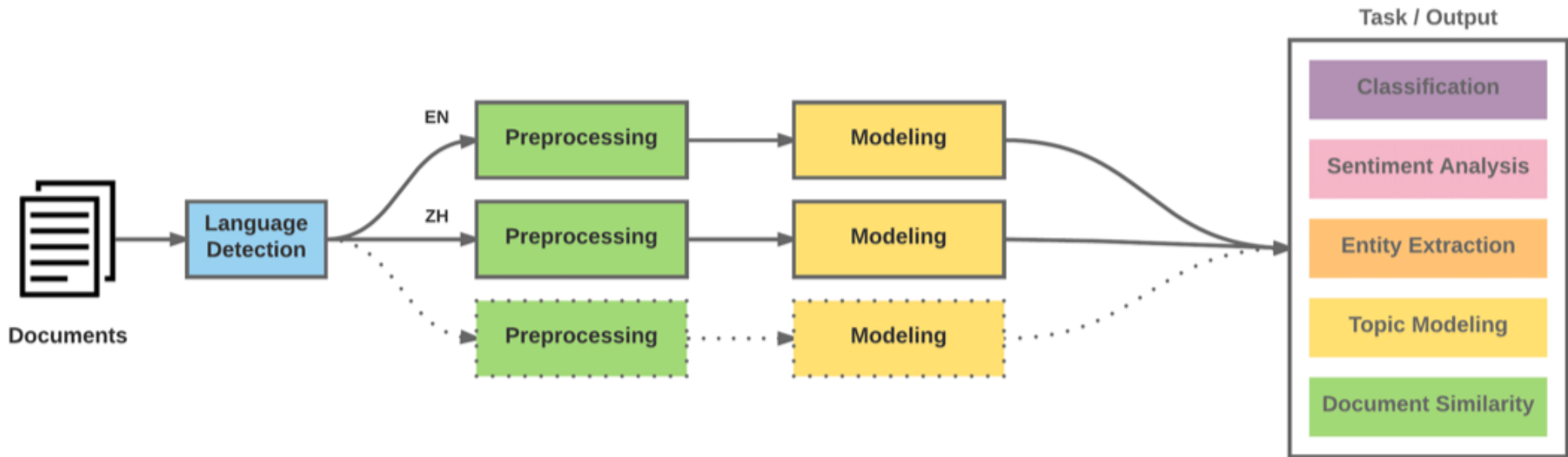
Deep Learning-based NLP



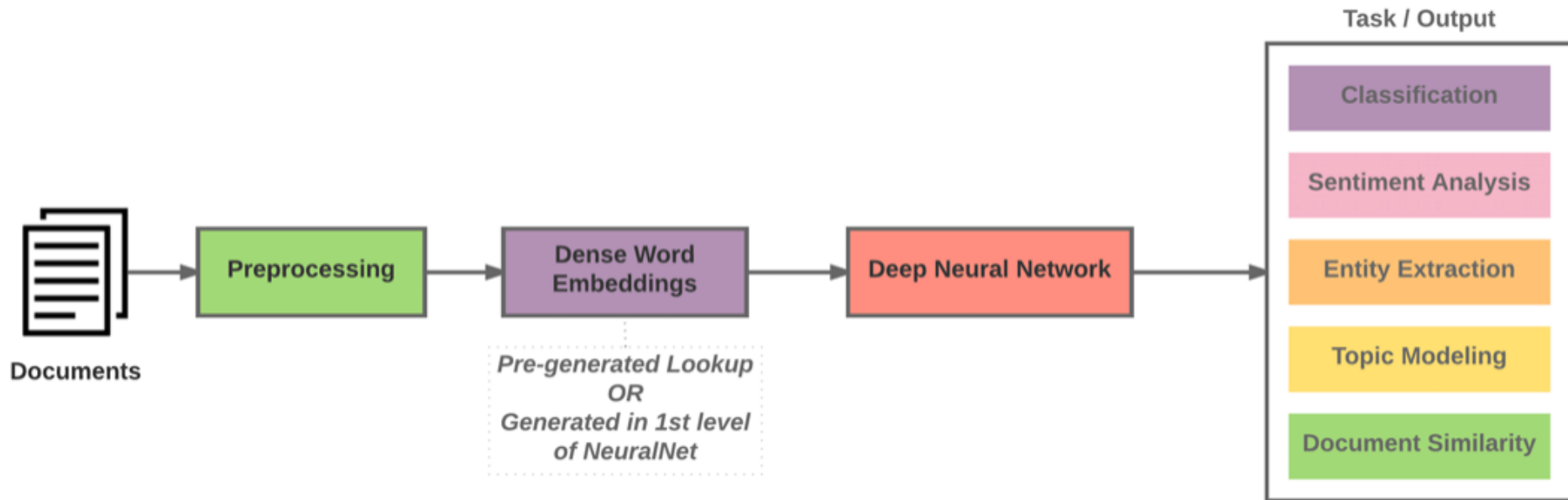
Modern NLP Pipeline



Modern NLP Pipeline



Deep Learning NLP



Natural Language Processing (NLP) and Text Mining

Raw text

Sentence Segmentation

Tokenization

Part-of-Speech (POS)

Stop word removal

Stemming / Lemmatization

Dependency Parser

String Metrics & Matching

word's stem

am → am

having → hav

word's lemma

am → be

having → have

Data Mining Tasks & Methods

Data Mining Tasks & Methods	Data Mining Algorithms	Learning Type
Prediction		
Classification	Decision Trees, Neural Networks, Support Vector Machines, kNN, Naïve Bayes, GA	Supervised
Regression	Linear/Nonlinear Regression, ANN, Regression Trees, SVM, kNN, GA	Supervised
Time series	Autoregressive Methods, Averaging Methods, Exponential Smoothing, ARIMA	Supervised
Association		
Market-basket	Apriori, OneR, ZeroR, Eclat, GA	Unsupervised
Link analysis	Expectation Maximization, Apriori Algorithm, Graph-Based Matching	Unsupervised
Sequence analysis	Apriori Algorithm, FP-Growth, Graph-Based Matching	Unsupervised
Segmentation		
Clustering	k-means, Expectation Maximization (EM)	Unsupervised
Outlier analysis	k-means, Expectation Maximization (EM)	Unsupervised

Example of Cluster Analysis

Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)

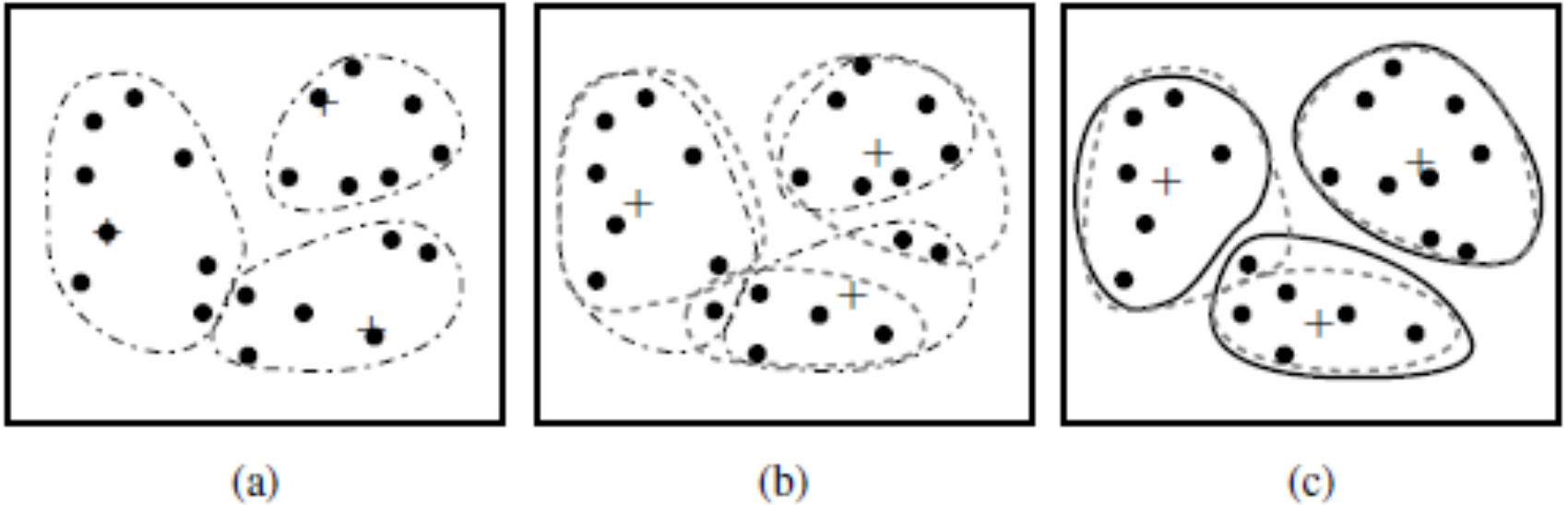
m2 (6.75, 3.50)

Cluster Analysis

Cluster Analysis

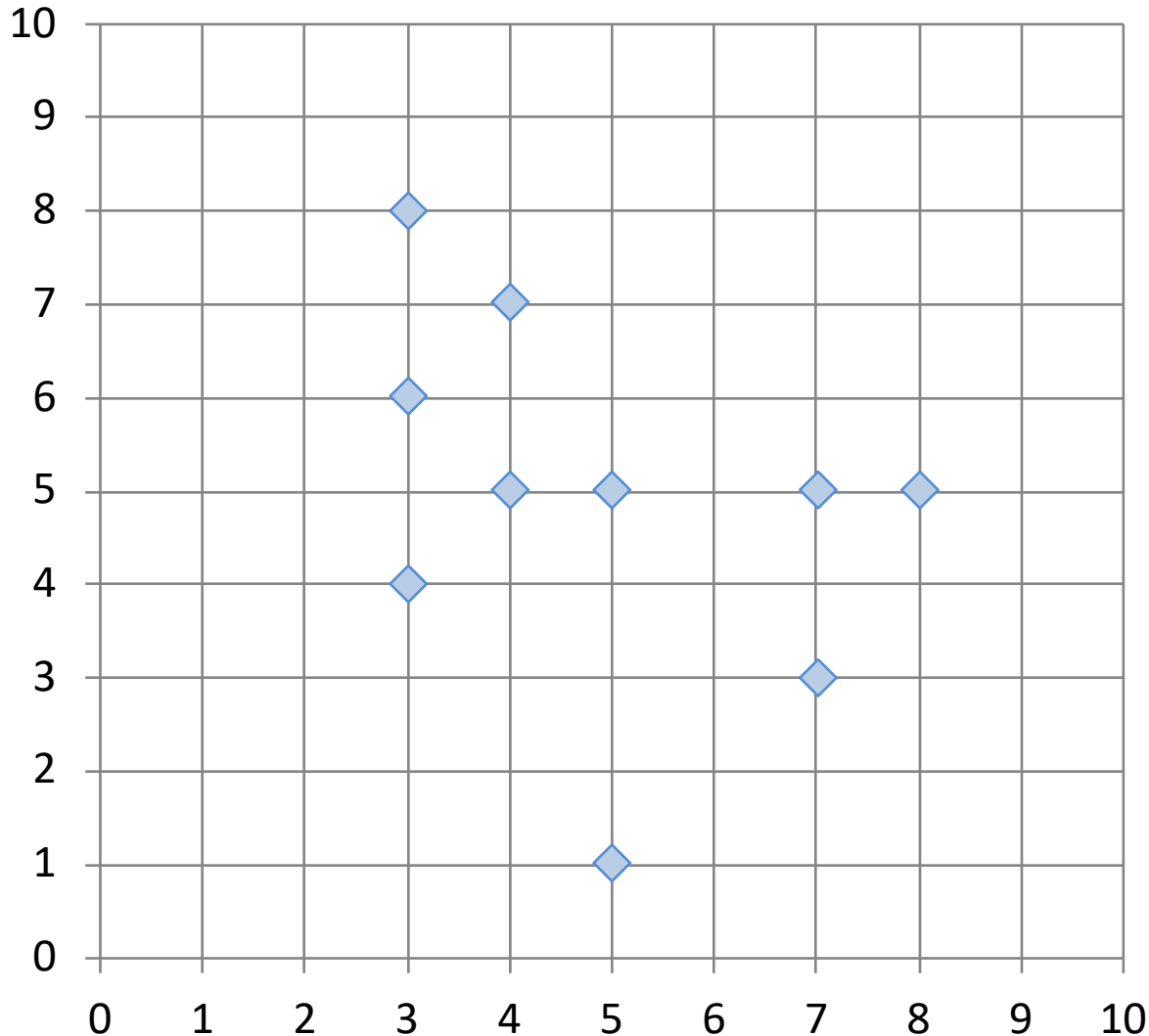
- Used for automatic identification of **natural groupings** of things
- Part of the machine-learning family
- Employ **unsupervised learning**
- Learns the clusters of things from past data, then assigns new instances
- There is not an output variable
- Also known as **segmentation**

Cluster Analysis



Clustering of a set of objects based on the *k-means method*.
(The mean of each cluster is marked by a “+”.)

Example of Cluster Analysis



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

Cluster Analysis for Data Mining

- **How many clusters?**
 - There is not a “truly optimal” way to calculate it
 - Heuristics are often used
 1. Look at the sparseness of clusters
 2. **Number of clusters = $(n/2)^{1/2}$** (n: no of data points)
 3. Use Akaike information criterion (AIC)
 4. Use Bayesian information criterion (BIC)
- Most cluster analysis methods involve the use of a **distance measure** to calculate the closeness between pairs of items
 - **Euclidian** versus **Manhattan** (rectilinear) **distance**

***k*-Means Clustering Algorithm**

- k : pre-determined number of clusters
- Algorithm (**Step 0**: determine value of k)

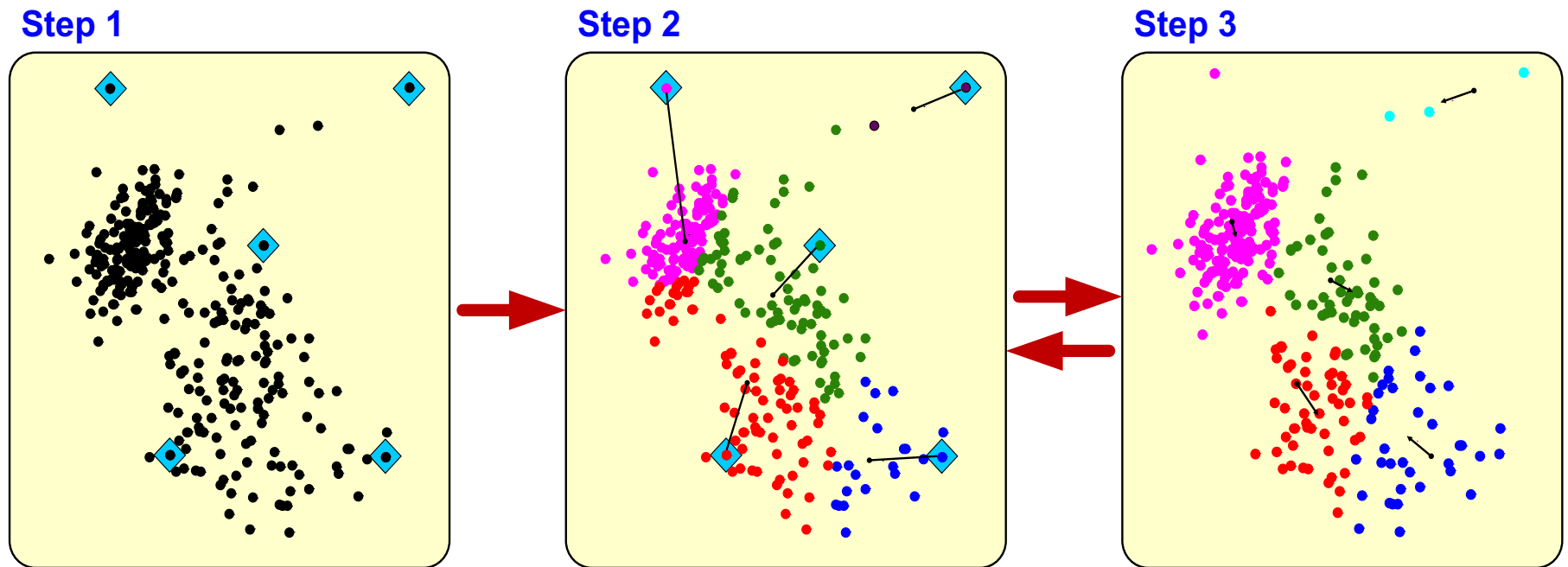
Step 1: Randomly generate k random points as initial cluster centers

Step 2: Assign each point to the nearest cluster center

Step 3: Re-compute the new cluster centers

Repetition step: Repeat steps 2 and 3 until some convergence criterion is met (usually that the assignment of points to clusters becomes stable)

Cluster Analysis for Data Mining - *k*-Means Clustering Algorithm



Similarity

Distance

Similarity and Dissimilarity Between Objects

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p -dimensional data objects, and q is a positive integer

- If $q = 1$, d is **Manhattan distance**

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

Similarity and Dissimilarity Between Objects (Cont.)

- If $q = 2$, d is **Euclidean distance**:

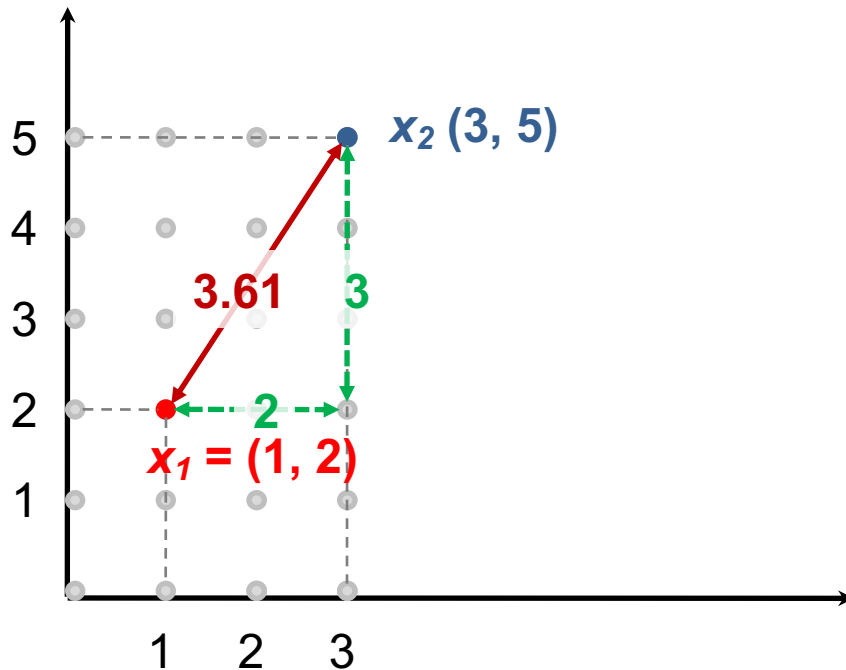
$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

– Properties

- $d(i,j) \geq 0$
 - $d(i,i) = 0$
 - $d(i,j) = d(j,i)$
 - $d(i,j) \leq d(i,k) + d(k,j)$
- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

Euclidean distance vs Manhattan distance

- Distance of two point $x_1 = (1, 2)$ and $x_2 (3, 5)$



Euclidean distance:

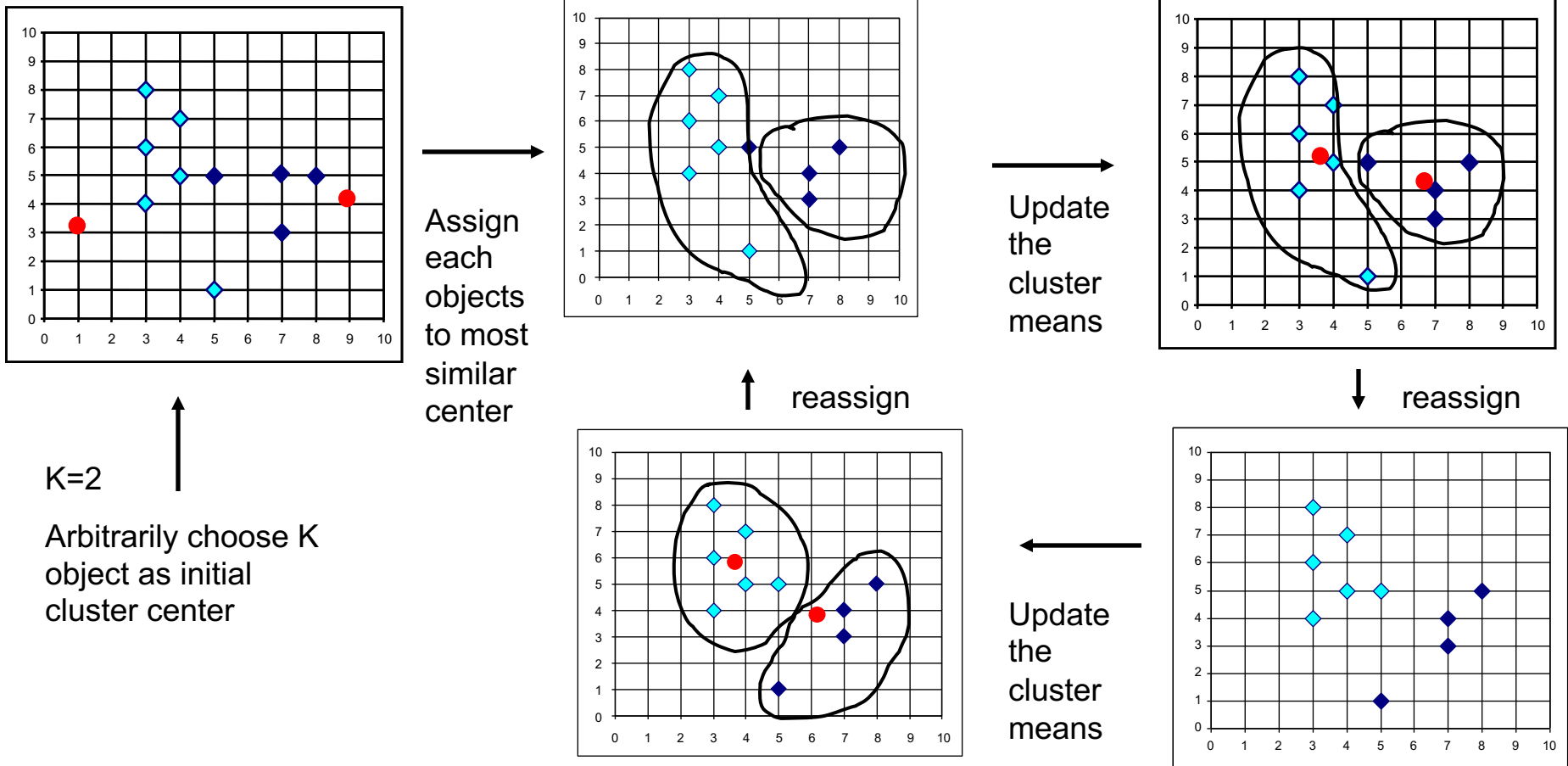
$$\begin{aligned} &= ((3-1)^2 + (5-2)^2)^{1/2} \\ &= (2^2 + 3^2)^{1/2} \\ &= (4 + 9)^{1/2} \\ &= (13)^{1/2} \\ &= 3.61 \end{aligned}$$

Manhattan distance:

$$\begin{aligned} &= (3-1) + (5-2) \\ &= 2 + 3 \\ &= 5 \end{aligned}$$

The *K-Means* Clustering Method

- Example



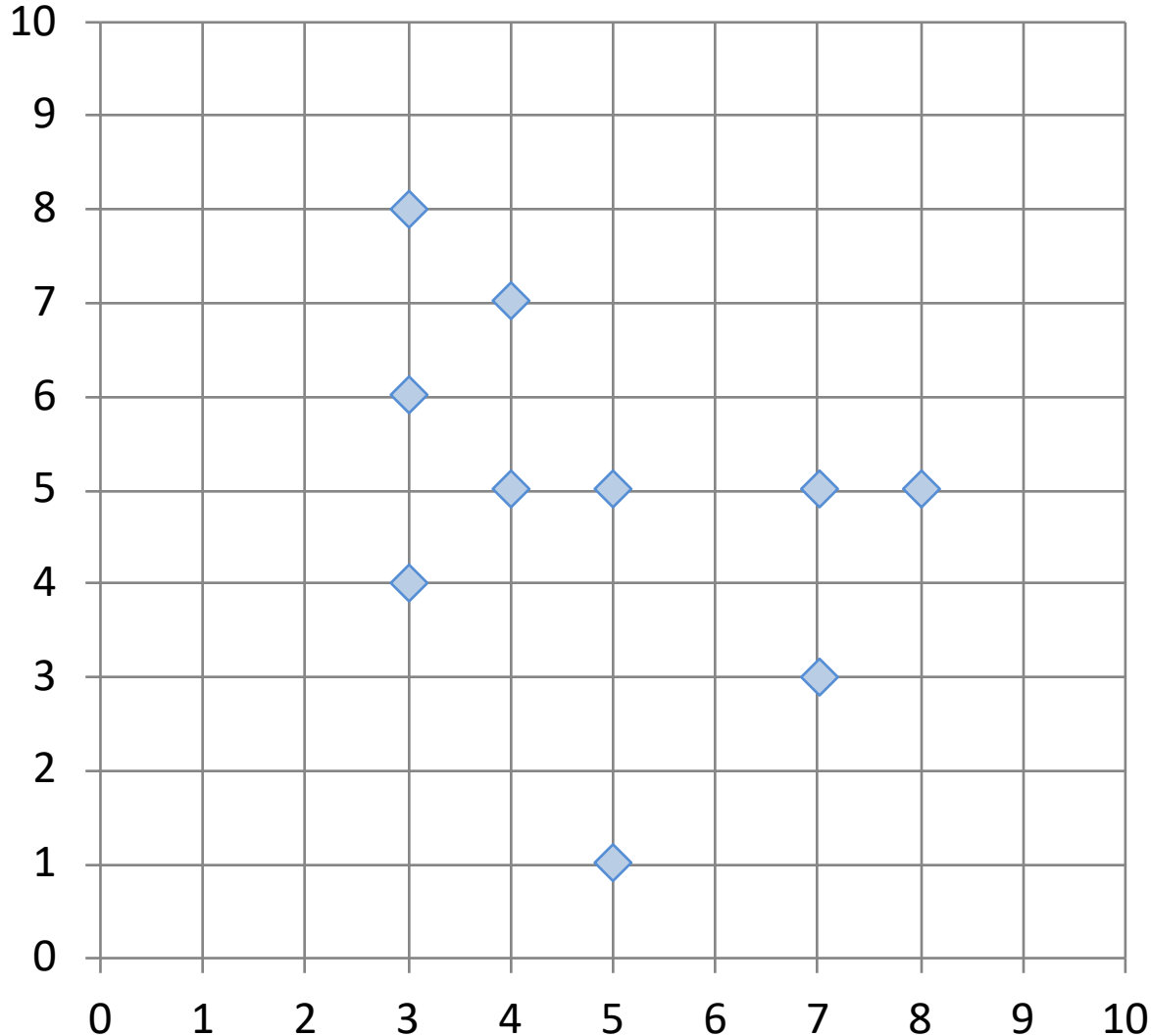
K-Means Clustering

Example of Cluster Analysis

Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

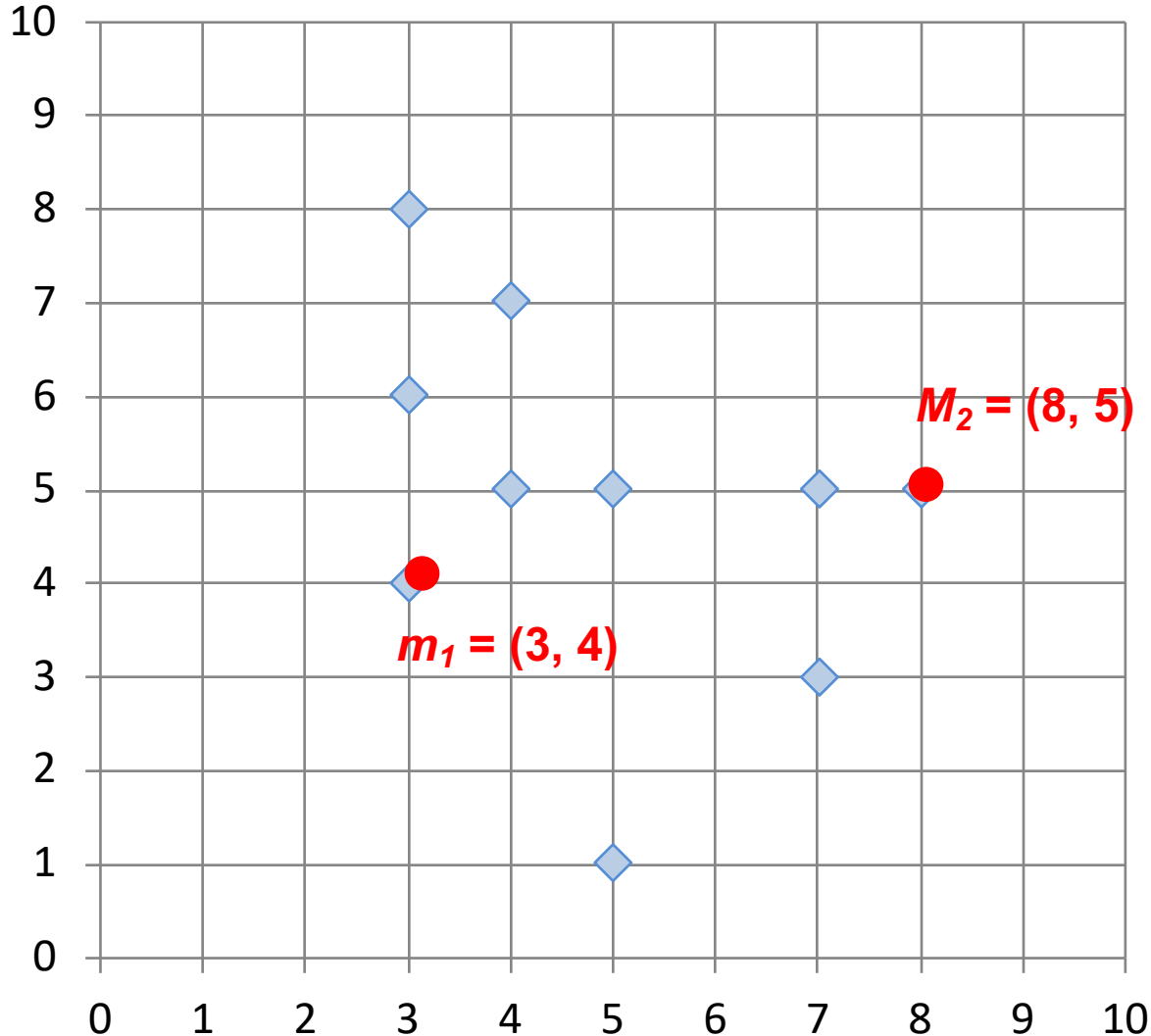
Step by Step



Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

K-Means Clustering

Step 1: K=2, Arbitrarily choose K object as initial cluster center

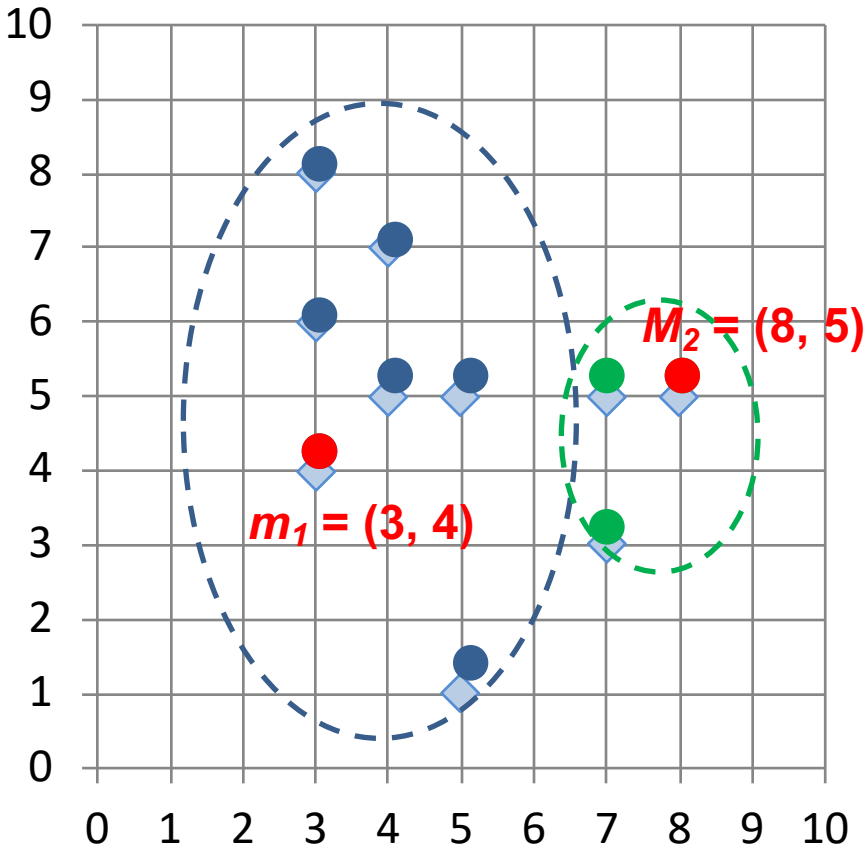


Point	P	P(x,y)
p01	a	(3, 4)
p02	b	(3, 6)
p03	c	(3, 8)
p04	d	(4, 5)
p05	e	(4, 7)
p06	f	(5, 1)
p07	g	(5, 5)
p08	h	(7, 3)
p09	i	(7, 5)
p10	j	(8, 5)

Initial m_1 (3, 4)
Initial m_2 (8, 5)

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center



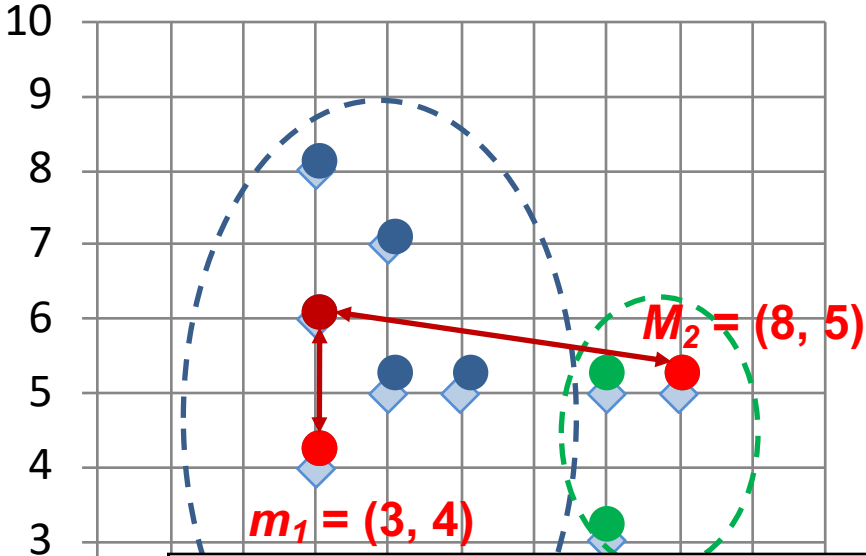
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1
p05	e	(4, 7)	3.16	4.47	Cluster1
p06	f	(5, 1)	3.61	5.00	Cluster1
p07	g	(5, 5)	2.24	3.00	Cluster1
p08	h	(7, 3)	4.12	2.24	Cluster2
p09	i	(7, 5)	4.12	1.00	Cluster2
p10	j	(8, 5)	5.10	0.00	Cluster2

Initial m1 (3, 4)
Initial m2 (8, 5)

K-Means Clustering

Step 2: Compute seed points as the centroids of the clusters of the current partition

Step 3: Assign each objects to most similar center



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	0.00	5.10	Cluster1
p02	b	(3, 6)	2.00	5.10	Cluster1
p03	c	(3, 8)	4.00	5.83	Cluster1
p04	d	(4, 5)	1.41	4.00	Cluster1

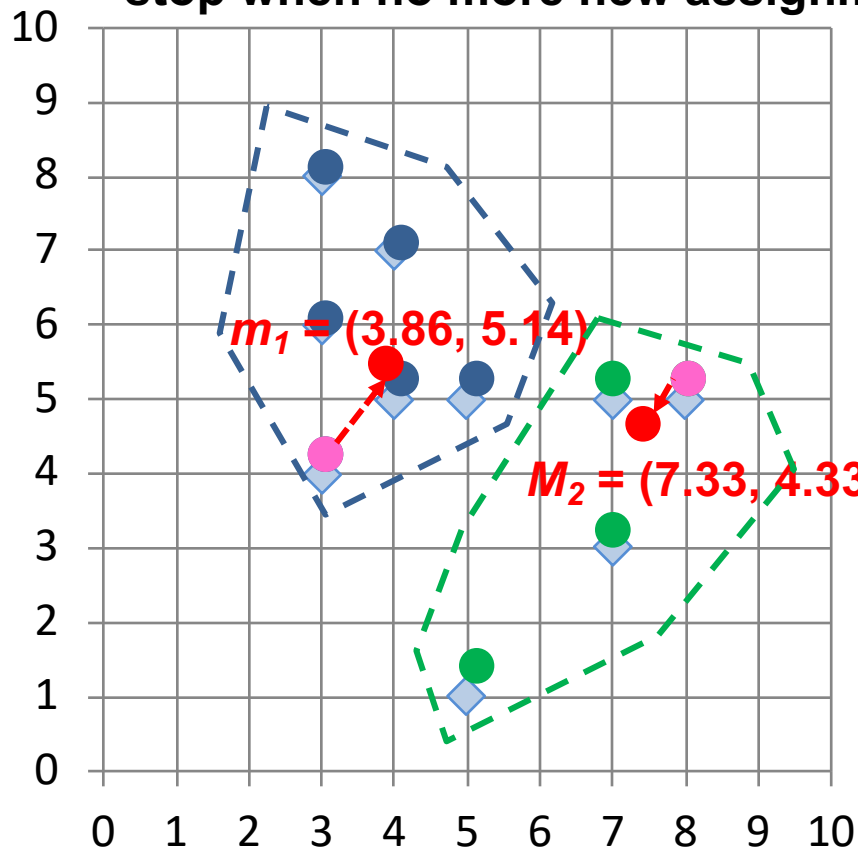
K-1

Euclidean distance
 $b(3,6) \leftrightarrow m1(3,4)$
 $= ((3-3)^2 + (4-6)^2)^{1/2}$
 $= (0^2 + (-2)^2)^{1/2}$
 $= (0 + 4)^{1/2}$
 $= (4)^{1/2}$
 $= 2.00$

Euclidean distance
 $b(3,6) \leftrightarrow m2(8,5)$
 $= ((8-3)^2 + (5-6)^2)^{1/2}$
 $= (5^2 + (-1)^2)^{1/2}$
 $= (25 + 1)^{1/2}$
 $= (26)^{1/2}$
 $= 5.10$

Initial m1 (3, 4)
 Initial m2 (8, 5)

**Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment**



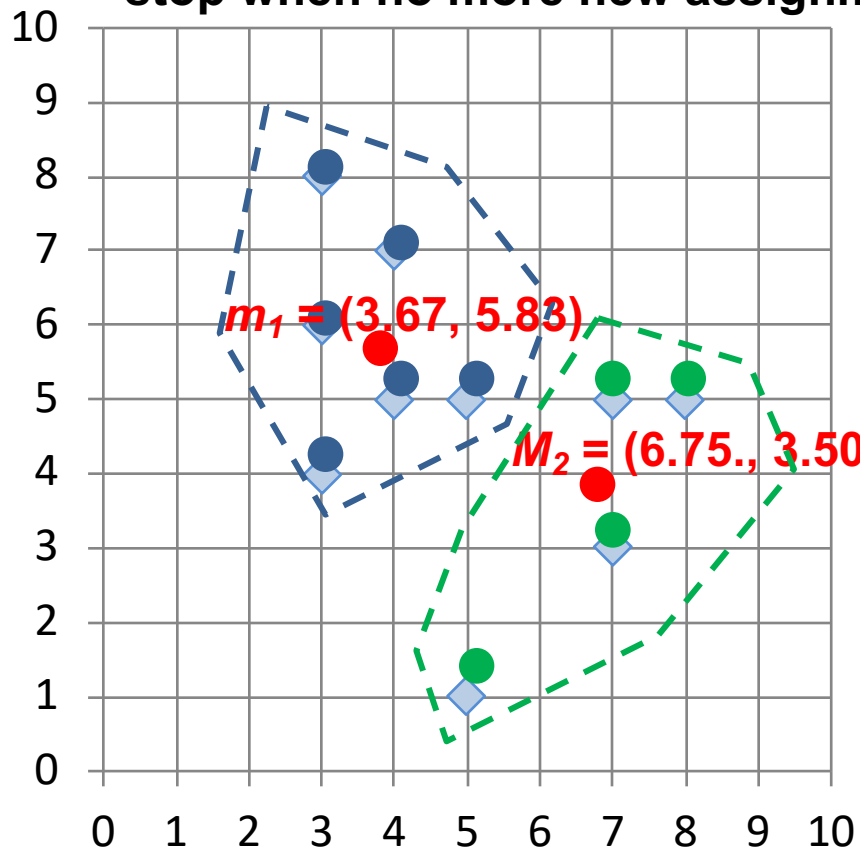
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.43	4.34	Cluster1
p02	b	(3, 6)	1.22	4.64	Cluster1
p03	c	(3, 8)	2.99	5.68	Cluster1
p04	d	(4, 5)	0.20	3.40	Cluster1
p05	e	(4, 7)	1.87	4.27	Cluster1
p06	f	(5, 1)	4.29	4.06	Cluster2
p07	g	(5, 5)	1.15	2.42	Cluster1
p08	h	(7, 3)	3.80	1.37	Cluster2
p09	i	(7, 5)	3.14	0.75	Cluster2
p10	j	(8, 5)	4.14	0.95	Cluster2

m_1 (3.86, 5.14)

m_2 (7.33, 4.33)

***K-Means* Clustering**

**Step 4: Update the cluster means,
Repeat Step 2, 3,
stop when no more new assignment**



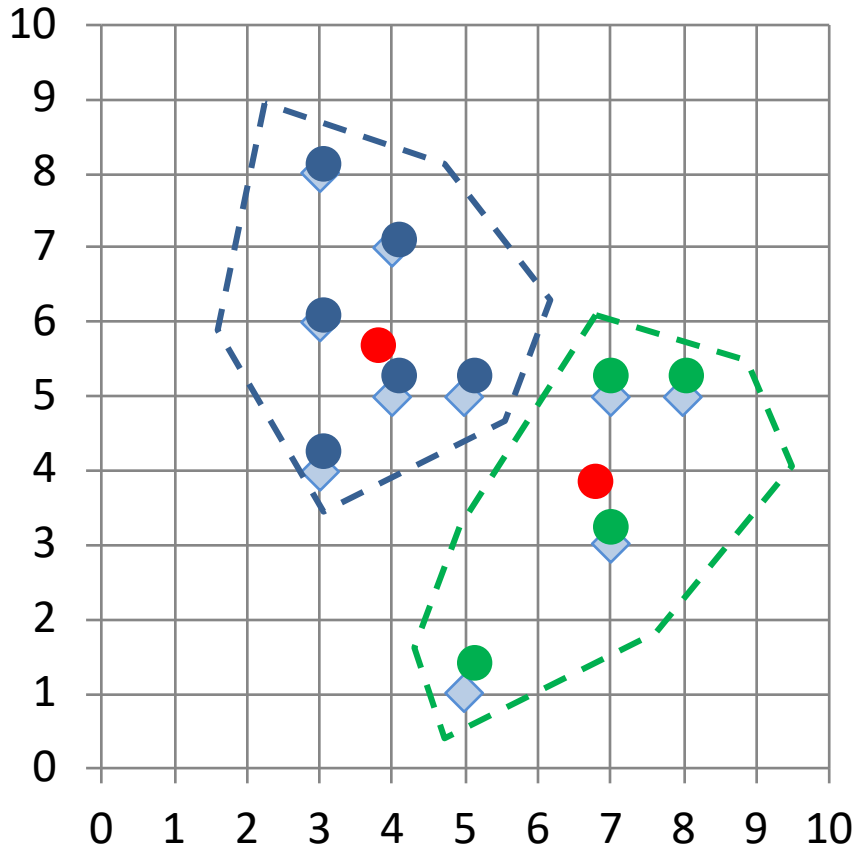
Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)

m2 (6.75, 3.50)

***K-Means* Clustering**

stop when no more new assignment



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

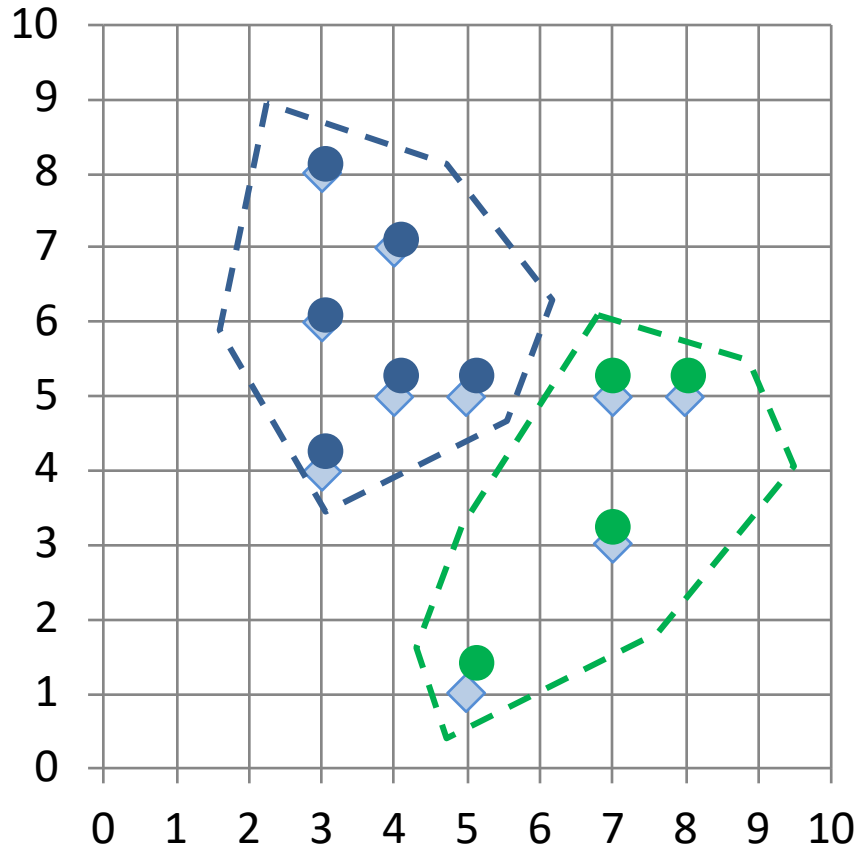
m1 (3.67, 5.83)

m2 (6.75, 3.50)

K-Means Clustering

K-Means Clustering ($K=2$, two clusters)

stop when no more new assignment



Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

K-Means Clustering

m1 (3.67, 5.83)

m2 (6.75, 3.50)

K-Means Clustering

Point	P	P(x,y)	m1 distance	m2 distance	Cluster
p01	a	(3, 4)	1.95	3.78	Cluster1
p02	b	(3, 6)	0.69	4.51	Cluster1
p03	c	(3, 8)	2.27	5.86	Cluster1
p04	d	(4, 5)	0.89	3.13	Cluster1
p05	e	(4, 7)	1.22	4.45	Cluster1
p06	f	(5, 1)	5.01	3.05	Cluster2
p07	g	(5, 5)	1.57	2.30	Cluster1
p08	h	(7, 3)	4.37	0.56	Cluster2
p09	i	(7, 5)	3.43	1.52	Cluster2
p10	j	(8, 5)	4.41	1.95	Cluster2

m1 (3.67, 5.83)

m2 (6.75, 3.50)

gensim

Fork me on GitHub



gensim

topic modelling for humans



Download

latest version from the Python Package Index



Direct install with:
easy_install -U gensim

Home

Tutorials

Install

Support

API

About

```
>>> from gensim import corpora, models, similarities
>>>
>>> # Load corpus iterator from a Matrix Market file on disk.
>>> corpus = corpora.MmCorpus('/path/to/corpus.mm')
>>>
>>> # Initialize Latent Semantic Indexing with 200 dimensions.
>>> lsi = models.LsiModel(corpus, num_topics=200)
>>>
>>> # Convert another corpus to the latent space and index it.
>>> index = similarities.MatrixSimilarity(lsi[another_corpus])
>>>
>>> # Compute similarity of a query vs. indexed documents
>>> sims = index[query]
```

Gensim is a FREE Python library



Scalable statistical semantics



Analyze plain-text documents for semantic structure



Retrieve semantically similar documents

spaCy

spaCy

HOME USAGE API DEMOS BLOG

Industrial-Strength Natural Language Processing in Python

Fastest in the world

spaCy excels at large-scale information extraction tasks. It's written from the ground up in carefully memory-managed Cython. Independent research has confirmed that spaCy is the fastest in the world. If your application needs to process entire web dumps, spaCy is the library you want to be using.

Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive. I like to think of spaCy as the Ruby on Rails of Natural Language Processing.

Deep learning

spaCy is the best way to prepare text for deep learning. It interoperates seamlessly with [TensorFlow](#), [Keras](#), [Scikit-Learn](#), [Gensim](#) and the rest of Python's awesome AI ecosystem. spaCy helps you connect the statistical models trained by these libraries to the rest of your application.

<https://spacy.io/>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>

python101.ipynb ☆

File Edit View Insert Runtime Tools Help [All changes saved](#)

Comment Share Settings A

RAM Disk Editing

Table of contents

- Build the model
- Train the model
- Evaluate the model
- Create a graph of accuracy and loss over time
- Text Classification: BBC News Articles
- Text Summarization and Topic Modeling
- Text Summarization
 - Text Summarization with Gensim Summarization
- Topic Modeling
 - Topic Modeling with Gensim LSI model
 - Topic Modeling with Gensim LDA model
 - Topic Modeling with Scikit-learn LDA and NMF
 - Topic Modeling Visualization
- Text Similarity and Clustering
 - Text Similarity**
 - Text Clustering
- Data Visualization
- Section

Text Similarity and Clustering

Text Similarity

- Spacy Vectors Similarity: <https://spacy.io/usage/vectors-similarity>

```
[1] 1 !python -m spacy download en_core_web_sm
```

```
[2] 1 !python -m spacy download en_core_web_lg
    2 # Restart Runtime
```

```
[3] 1 import spacy
    2 nlp = spacy.load("en_core_web_lg")
    3 tokens = nlp("apple banana cat dog notaword")
    4 for token in tokens:
    5     print(token.text, token.has_vector, token.vector_norm, token.is_oov)
```

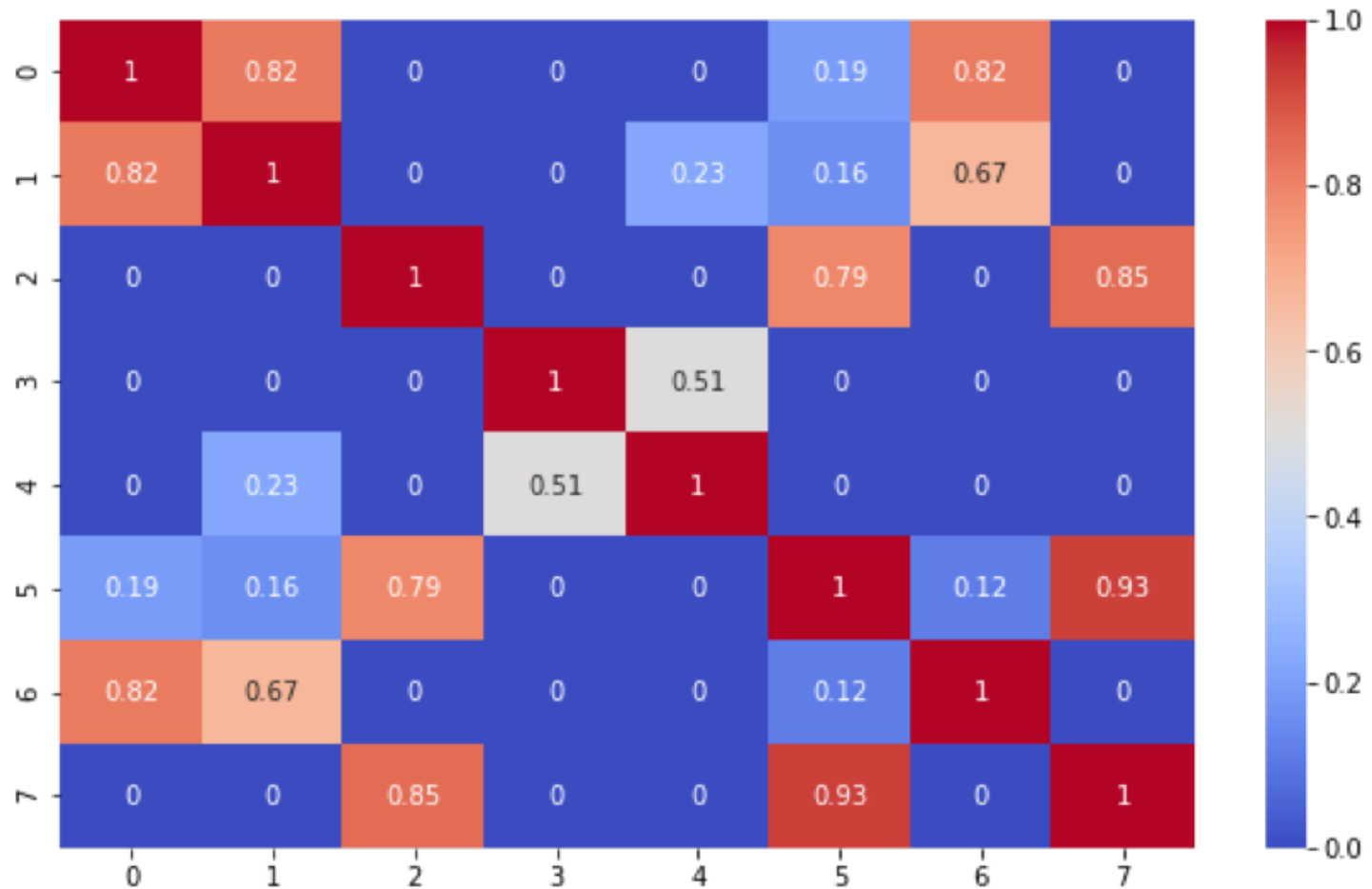
```
apple True 7.1346846 False
banana True 6.700014 False
cat True 6.6808186 False
dog True 7.0336733 False
notaword False 0.0 True
```

```
1 import spacy
2 nlp = spacy.load("en_core_web_lg")
3 doc1 = nlp("I like cat.")
4 doc2 = nlp("I like dog.")
5 doc1.similarity(doc2)
```

<https://tinyurl.com/aintpuppython101>

Python in Google Colab (Python101)

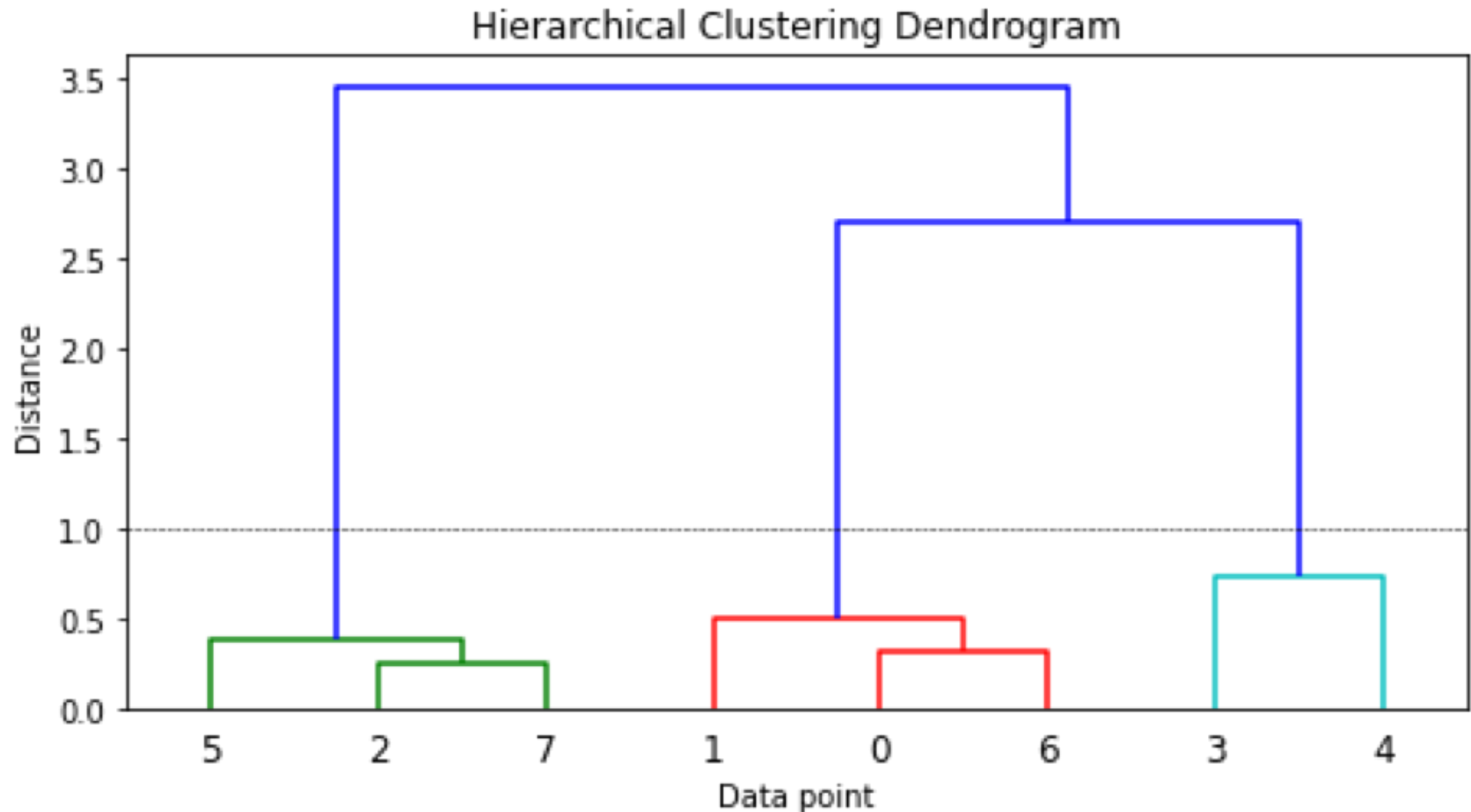
<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>



<https://tinyurl.com/aintpuython101>

Python in Google Colab (Python101)

<https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT>



<https://tinyurl.com/aintpupython101>

Summary

- Text Similarity
- Text Clustering
 - Cluster Analysis
 - K-Means Clustering

References

- Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress. <https://github.com/Apress/text-analytics-w-python-2e>
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python, O'Reilly Media. <https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/>
- Min-Yuh Day (2020), Python 101, <https://tinyurl.com/aintpupython101>